

AMENDMENTS TO THE SPECIFICATION

On page 6, please amend the paragraph starting on line 11 as follows:

~~Fig. 2 illustrates~~ Figs. 2A-2C, hereinafter collectively referred to as Fig. 2,
illustrate an exemplary network of the invention.

On page 6, please amend the paragraph starting on line 20 as follows:

~~Fig. 7 is~~ Figs. 7A-7D, hereinafter collectively referred to as Fig. 7, are an
exemplary flow diagram of a method of using the system of the invention.

On page 7, please amend the paragraph starting on line 4 as follows:

~~Fig. 10 illustrates~~ Figs. 10A-10C, hereinafter collectively referred to as FIG. 10,
illustrate a flow diagram of an embodiment of the graphical user interface of the
invention.

On page 7, please amend the paragraph starting on line 20 as follows:

Fig. ~~17R~~ 17S depicts an illustrative screen shot of an address book entry screen.

On page 8, please amend the paragraph starting on line 1 as follows:

Fig. ~~17S~~ 17R depicts an illustrative screen shot of an account file screen that
~~allow~~ allows a user to verify and modify account settings such as password, account
address, and default delivery.

On page 9, please amend the paragraph starting on line 10 as follows:

The network 300 can be divided, for explanatory purposes, into three sections: the client side of the network 300a, the back end side of the network 300b and the printing side of the network 300c. Communication between the client side 300a, the back end side 300b and the printer side 300c occurs, for example, through the network 300. As illustrated in Fig. 2, added levels of security, such as use of firewalls, ensure that information sent over the network 300 is not disturbed (e.g. the information is not modified, changed or breached). Generally speaking, a user on the client side 300a of the network can request printing from, for example, a personal computer, and generate a document for shipping and/or delivery from printer side 300c. The back end side 300b of the network ~~300b~~ 300 is transparent to the user.

On page 9, please amend the paragraph starting on line 20 and continuing to the following page as follows:

Fig. 3 is an exemplary diagram of an embodiment of the system in the present invention. Fig. 3 primarily illustrates the client side 300a of the network 300, with the addition of the UPJA 320 and database, such as a SQL server 332, which are on the back end side 300b of the network 300 as shown in Fig. 2. The system includes, for example, a Data Store, a Print Driver UI 310b, a local application 310a, a graphics library, a Print Driver PDL Gen, a Print Spooler 344, a Port Monitor 346, an Upload Manager 310c, a Universal Print Job Acceptor (UPJA) 320, a SQL server 332, a Version Manager 348, and a System Tray Application which allows pop-up menus and user selection of icons.

On page 10, please amend the paragraph starting on line 7 and continuing to the following page as follows:

First, relevant file information (parameters) will be sent from the spooler 344 via the port monitor 346 and upload manager ~~340a~~ 310c to a web server 320 (e.g. UPJA). The relevant file information includes, for example, a handle identifying the location of the printer, printer name, job id, printing level, and document information such as color, stapling, etc. After the relevant file information is sent to the port monitor 346, a check is performed to determine whether a valid component (e.g. a print driver 310b) is being used, for example, to print the document. In the preferred embodiment, the print driver resides as a file on the client, and comprises Windows™ based code. The print driver 310b may be installed, for example, by downloading it from a server or installed from disk. In order to validate the print driver 310b, the identity of the print driver 310b is sent to the version ~~manger~~ manager 348, and a check (e.g. a CRC) is performed to compare the components of the print driver stored in the file with information stored in a registry. The registry includes, for example, predetermined information which can authenticate that a valid print driver is being used. If the comparison results in an invalid print driver, the data file (document) will not be sent (uploaded) to the web server. If, on the other hand, the comparison results in a determination that the component (print driver) is valid, then the data file is sent from the print spooler 344 to the port monitor 346. The data file is sent, for example, as packets of information from the print spooler 344 to the port monitor 346. In this embodiment, it is preferable that the packets of data are sent in 4 Kbyte packets. Once the data file (either as a whole or as individual packets) has been sent to the port monitor 346, the data file or packet is sent, via the upload

manager 310c, to the UPJA 320 and stored. The upload manager 310c then launches a web browser 310d for viewing the document.

On page 14, please amend the paragraph starting on line 20 and continuing to the following page as follows:

Referring to Fig. 4, a user requests data to be printed at 400. After data (i.e. a document) has been selected at 400, the user may configure the settings of the document using a configuration unit (e.g. the item configuration wizard), as described below. Before uploading the selected document(s) to the UPJA 320, the print driver 310b selected by the user is verified by the version manager 348 at 410. The version manager is preferably a separate component that is accessed to verify that all of the client components being used by the client are valid before any uploading occurs. If the version manager 348 determines that the print driver is not valid (i.e. not acceptable), then a message is sent to the client 310 at 450. If, on the other hand, the version manager 348 determines that the print driver information is valid, then data is uploaded from the client 310, for example, to the port monitor 346 via the print spooler 344 (see, for example, Fig. 37 3). That is, when data being uploaded includes version information, the upload manager 310c can query about, for example, what version of software the system supports. For example, when a "print" command is issued, the upload manager 310c can monitor, in real time, the data to determine if a valid version of the print driver 310b (e.g. driver software) is being used. The upload manager 310c also verifies that the client 310 software (e.g. local applications), and/or components making up the software, represent a coherent set of components and work properly

together. This can be accomplished, for example, by a version manager 348 authenticating the components of the drivers and client software.

On page 16, please amend the paragraph starting on line 7 as follows:

As illustrated in Fig. 5, users can go to a website (400a) and upload/download application data (420a). ~~subsequent~~ Subsequent to uploading a document in 420 420a, in one embodiment users can specify with the aid of a configuration wizard, the item, for example, document type, paper stock, printing options, double or single sided copies, color versus black and white, cover and orientation (422). The user is then presented with the option of either viewing their shopping cart and choosing an item (e.g. another document) to be purchased for printing, or can select the number of copies to be printed (422). After the selection of the number of copies has been made, the user can select the appropriate address or location for delivery of the printed document, convey billing information, preview and submit the order, and receive confirmation and updates regarding the order (424 and 442).

On page 17, please amend the paragraph starting on line 10 and continuing to the following page as follows:

Fig. 7 is an exemplary flow diagram of a method of using the system in the invention. The flow illustrated is merely an example of one embodiment of a process that a user accessing the system may perform. The user creates a document in a local application 310a on, for example, their personal computer (600), and prints the documents from the application to the selected print driver 310b (605). The document

is printed as a postscript document (610). The upload manager 310c then uploads the postscript print file to the UPJA 320 over http/https (615), and the upload manager launches the web browser 310d for document viewing (620). The user can then configure the finishing and binding options for the document using the interface on the personal computer (625). Once document configuration information is validated, the user inputs shipping and payment data on the interface (630). The shipping and payment data are verified, and the print file is put in long term storage (635). The finishing and ~~binding~~ binding options are then combined with the postscript file to create a print ready file (640), and the print ready file is sent to the print queue (645) and transferred to the production facility (i.e. printing facility) (650). A printer operator can then select a job and queues it to an available printer (655), and the job is ripped and sent to the printer (660). The printer punches and/or binds the job on-line (665), the printed document is shrink wrapped and packaged (670), and the package is sent for delivery (675). The user is able to track the package, receive the package and open the package (680-690).

On page 26, please amend the paragraph starting on line 4 as follows:

Fig. 47R 17S depicts an illustrative screen shot of an address book entry screen.

On page 26, please amend the paragraph starting on line 5 as follows:

Fig. 47S 17R depicts an illustrative screen shot of an account file screen that ~~allow~~ allows a user to verify and modify account settings such as password, account address, and default delivery.